

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



SAYISAL İŞARET İŞLEME YÖNTEMLERİ İLE MÜZİK
TÜRÜ SINIFLANDIRMA

17011057 – Tarık AYTEK
18011100 – Tuğrul Alp ÖZBUCAK

BİLGİSAYAR PROJESİ

Danışman
Öğr. Gör. Dr. Ahmet ELBİR

Mayıs, 2022

TEŞEKKÜR

Çalışmamızın her adımında konu üzerindeki engin bilgileriyle bize yol gösteren danışman hocamız Öğr. Gör. Dr. Ahmet ELBİR'e en derin teşekkürlerimizi sunuyoruz. Kendilerinin gözetimi ve tavsiyeleri olmadan projenin hedefine ulaşması oldukça zor olurdu.

Ayrıca bu araştırma konusu hakkında bizimle aynı anda çalışan, bölüm arkadaşlarımız Berk Sıcak ve Aytuğ İlayda Baldede'ye eksik kaldığımız yerlerde ilham verdikleri için teşekkür ederiz.

Tarık AYTEK
Tuğrul Alp ÖZBUCAK

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
ÖZET	viii
ABSTRACT	ix
1 Giriş	1
1.1 Çalışmanın Amacı ve Motivasyonu	1
2 Ön İnceleme	2
2.1 Yapılmış Çalışmalar	2
2.2 Değerlendirme	3
3 Fizibilite	4
3.1 Teknik Fizibilite	4
3.1.1 Yazılım Fizibilitesi	4
3.1.2 Donanım Fizibilitesi	4
3.2 İş Gücü ve Zaman Planlaması	5
3.3 Yasal Fizibilite	5
3.4 Ekonomik Fizibilite	5
4 Sistem Analizi	6
4.1 Çalışmanın Hedefleri	6
4.2 Gereksinim Analizi	6
4.3 Performans Metrikleri	7
5 Sistem Tasarımı	8
5.1 Ses Dosyalarının Hazırlanması	8
5.2 Öz Niteliklerin Elde Edilmesi	9
5.3 Kümeleme Algoritmaları	9

5.4 Modelleme Algoritmaları	10
6 Uygulama	12
7 Deneysel Sonuçlar	13
8 Performans Analizi	18
9 Sonuç	19
Referanslar	20
Özgeçmiş	22

KISALTMA LİSTESİ

COG	Center of Gravity
CQT	Constant Q Transform
DFT	Discrete Fourier Transform
DWCH	Daubechies Wavelet Coefficient Histograms
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDE	Integrated Development Environment
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coding
MFCC	Mel-frequency Cepstral Coefficients
MGR	Music Genre Recognition
MIR	Music Information Retrieval
NN	Nearest Neighbor
STFT	Short Time Fourier Transform
SVM	Support Vector Machine

ŞEKİL LİSTESİ

Şekil 3.1	İş Gücü ve Zaman Planlaması	5
Şekil 5.1	Sistem Çalışma Akışı	8
Şekil 5.2	Ses dosyalarını bölme işlemi	8
Şekil 6.1	Kümelenmemiş veri setinden örnek kesit	12
Şekil 6.2	GMM algoritması ile vektörleri 3 kümeye ayrılmış veri setinden örnek kesit	12

TABLO LİSTESİ

Tablo 2.1	Kümeleme Sonuçları [1]	2
Tablo 7.1	Herhangi bir kümeleme uygulanmadan elde edilen sonuçlar . .	13
Tablo 7.2	K-Means Clustering algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar	14
Tablo 7.3	K-Means Clustering algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar	14
Tablo 7.4	GMM algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar	15
Tablo 7.5	GMM algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar	15
Tablo 7.6	Hierarchical Clustering algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar	16
Tablo 7.7	Hierarchical Clustering algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar	16
Tablo 7.8	Genel karşılaştırma tablosu	17

SAYISAL İŞARET İŞLEME YÖNTEMLERİ İLE MÜZİK TÜRÜ SINIFLANDIRMA

Tarık AYTEK

Tuğrul Alp ÖZBUCAK

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Öğr. Gör. Dr. Ahmet ELBİR

Müzik Türü Sınıflandırma hala mükemmelleştirilmemiş bir alandır. Sınıflandırma başarısını artırabilmek adına devamlı yeni çalışmalar yapılmaktadır. Bu çalışma, ses sinyallerinin öznitelik vektörleri üzerinde uygulanan çeşitli kümeleme algoritmalarının, sınıflandırma başarı oranlarını nasıl etkilediğini göstermektedir.

GTZAN veri setindeki müzik dosyalarından sinyal işleme teknikleri ile elde edilen öznitelik vektörleri, çeşitli kümeleme algoritmaları ile kümelenir. Farklı kümelendirmelerden oluşan veri setleri üzerinde çeşitli makine öğrenmesi metotları uygulanır ve sonuçlar farklı metrikler aracılığıyla karşılaştırılır. Karşılaştırmalar sonucunda kümeleme yöntemi ve sınıflandırma modeli farketmeksizin, öznitelik vektörlerini kümelemenin sınıflandırma başarısını düşürdüğü görülmüştür.

Anahtar Kelimeler: Müzik Türü Sınıflandırma, Gözetimsiz Kümeleme, Makine Öğrenmesi, Sinyal İşleme

ABSTRACT

MUSIC GENRE CLASSIFICATION WITH DIGITAL SIGNAL PROCESSING METHODS

Tarık AYTEK
Tuğrul Alp ÖZBUCAK

Department of Computer Engineering
Computer Project

Advisor: Assoc. Prof. Dr. Ahmet ELBİR

Music Genre Classification is a field that's still not perfected. In order to increase the classification success, new studies are carried out continuously. This study shows how various clustering algorithms applied on feature vectors of audio signals affect classification success rates.

The feature vectors obtained by signal processing techniques from the music files in the GTZAN dataset are clustered with various clustering algorithms. Various machine learning methods are applied on data sets consisting of different clusters and the results are compared through different metrics. As a result of the comparisons, it has been seen that clustering feature vectors reduces the classification success, regardless of the clustering method and classification model.

Keywords: Music Genre Classification, Unsupervised Clustering, Machine Learning, Signal Processing

Bu bölümde, çalışma tanıtılacak; amacı, hedefi, motivasyonu ortaya konacak ve hakkında genel bilgiler verilecektir.

1.1 Çalışmanın Amacı ve Motivasyonu

Gün geçtikçe sayıları hızla artan müzik parçaları, bilişim dünyasındaki müzik alanında çeşitli uygulamaların gelişmesine önayak olmuştur. Genelde tür sınıflandırma ve öneri algoritmalarına katkı sunan bu uygulamalar sinyal işleme, doğal dil işleme, makine öğrenmesi gibi alanların desteğiyle ileri taşınmaktadır. Bu çalışmanın amacı ise sinyal işleme ile müzik türü sınıflandırması yaparken ses özelliklerinin kümelendirilmesinin başarı yüzdesine katkısını ölçmektir. Bu çalışma ile bundan sonraki yapılacak sınıflandırma ve öneri çalışmalarına önayak olabilmek, çalışmayı farklı parametreler üzerinde ilerletebilmek, çalışmalar arası yapılabilecek karşılaştırmalara bir katkı sağlamış olabilmek motivasyonlarımız arasındadır.

Türlerine göre ayrılmış müzik dosyalarının bulunduğu veri kümemizdeki parçaların librosa kütüphanesi yardımı ile özellik çıkarımını yapılır. Özellik vektörlerinin üzerinde çeşitli kümeleme yöntemleri uygulanarak özelliklerimizin sayısı benzer özellikler bir arada bulunacak şekilde arttırılır. Bu yöntem 1 küme, 2 küme ve 3 küme kullanılması ile ayrı ayrı kümeleme yöntemleri ile uygulandıktan sonra başarı oranlarının karşılaştırması yapılır ve Gözetimsiz Kümeleme Yöntemlerinin fayda sağlayıp sağlamadığı görülecektir.

2.1 Yapılmış Çalışmalar

- Yapılan bir çalışmada öznitelik olarak tempo spektrumu, sıfır geçiş oranı, MFCC, LPC-türetilmiş septum, güç spektrumu kullanılmıştır. Kümeleme metodu olarak ise NN, GMM, HMM ve çalışmanın ana konusu olan SVM kullanılmıştır [1]. Sonuçlar aşağıdaki gibidir. Başarı oranı çok yüksek olsa da kıssas olarak alınan öznitelik miktarı biraz az gibi bulunmaktadır.

Tablo 2.1 Kümeleme Sonuçları [1]

	SVM	NN	GMM	HMM
Error Rate	6.86%	20.57%	12.31%	11.94%

- MIR konusunda en çok atıf yapılmış olan bu çalışmada ise tını doku özelliklerinden Spektral Merkez, Spektral Devrilme, Spektral Akış, Zaman Alan Sıfır Geçişleri, MFCC, Analiz ve Doku Pencerelemleri, Düşük Enerji Özelliği kullanılmış ve bu özellikleri ortalaması ve varyansı bir vektör halinde kullanılmıştır. Buna ek olarak ritmik ve perde özellikleri de karşılaştırma için kullanılmıştır. Sınıflandırma için ise HMM ve KNN kullanılmıştır [2]. Çalışmada GMM metoduyla Klasik müzikte 2 kümelemeyle %81'e yakın başarı alınırken 3 kümelemeden itibaren başarı oranının %7 arttığı gözlemlenmiştir. KNN ile ise başarı oranları küme miktarını arttırdıkça düşmüştür. Çalışmanın kendisinin daha kapsamlı olduğu göz önünde bulundurulmalıdır. İçerisinde insan tabanlı müzik türü tespiti oranı, hangi müzik türlerinin hangileriyle karıştırıldığı vb. gibi daha farklı konulara da değinilmektedir. Bu çalışmanın eksiği olarak olabildiğince çok öznitelik almasına karşın yine de düşük başarı oranı gösterilebilir. Aynı zamanda kümeleme yöntemleri çok kısıtlı tutulmuştur.
- Başka bir çalışmada ise tını doku özellikleri olarak: MFCC, Spektral Merkez, Spektral Devrilme, Spektral Akış, Sıfır Geçiş ve Düşük Enerji kullanılmıştır. Ritmik İçerik özellikleri olarak tempo, ritim ve zaman ölçüsü alınmıştır.

Perde özellikleri de kullanılmıştır. Kümeleme metotları olarak ise SVM, KNN, GMM, LDA kullanılmıştır. Ama çalışmanın asıl noktası öznitelik olarak alınan DWCH'lerdir. Hemen önceki çalışmaya işaret ederek, başarı oranının sadece %61'e çıkmasını bir eksik olarak gördüğünden bu yeni yolu seçtiklerini belirtmişlerdir. Başarı oranlarının genel olarak %80'e, spesifik türlerde ise %98'lere kadar çıktığı gözlemlenmiştir [3]. DWCH'lerde SVM2, SVM1'e kıyasla %3.6 daha başarı göstermiştir.

- GTZAN veri seti kullanan ve neredeyse bütün çalışmalarla kendini kıyaslayan bu örnekte ise önceki belirtilen bütün nitelik ve sınıflandırma metotlarının üstüne çeşitli daha önce kullanılmamış timbre ve temporal öznitelikler kullanılmıştır. Bu verilere Ortalama ve Varyans kombinasyonu ile MuVar denilen standart; ortalama ve kovaryans matrixinin üst üçgeninin birleştirilmesi kullanarak ise MuCov denilen standart elde edilmiştir [4]. SRC ve SVM kullanılarak başarı oranının %90 üzerinde olduğu görülmüştür. Tekrardan çalışmanın kendisinin çok daha derin olduğu akılda bulundurulmalıdır.

2.2 Değerlendirme

Bir önceki bölümde yapılmış çalışmalar ve daha da fazlası incelendiğinde görülmüştür ki, MIR hala mükemmelleştirilmemiş bir alandır. Geçmişte kullanılmış bazı tekniklere çok bağlı kalındığından başarı oranları düşük de çıkabilmektedir. Ama aynı zamanda yeni geliştirilen yöntemler ve bu yeni yöntemlerin kombinasyonları ile daha önce mümkün olmayan miktarda başarı oranlarına da erişebilmektedir.

Ancak araştırılmalardan görüldüğü üzere çıkarılmış olan öznitelikler çoğu zaman sadece ortalama ve varyans değerleri üzerinden değerlendirilmektedir. Sınıflandırma yöntemlerinde ise sadece Gözetimli Öğrenme yöntemlerinin kullanıldığı ve Gözetimsiz Öğrenme yöntemlerinin dışarıda bırakıldığı gözlemlenmiştir. Perde ve tempo özelliklerinin kendi başlarına neredeyse hiç faydası olmadığı da gözlemlenmiştir. Başarının en temel kısmı düşük seviye öznitelikleri olarak geçen tını özelliklerinden gelmektedir.

3.1 Teknik Fizibilite

3.1.1 Yazılım Fizibilitesi

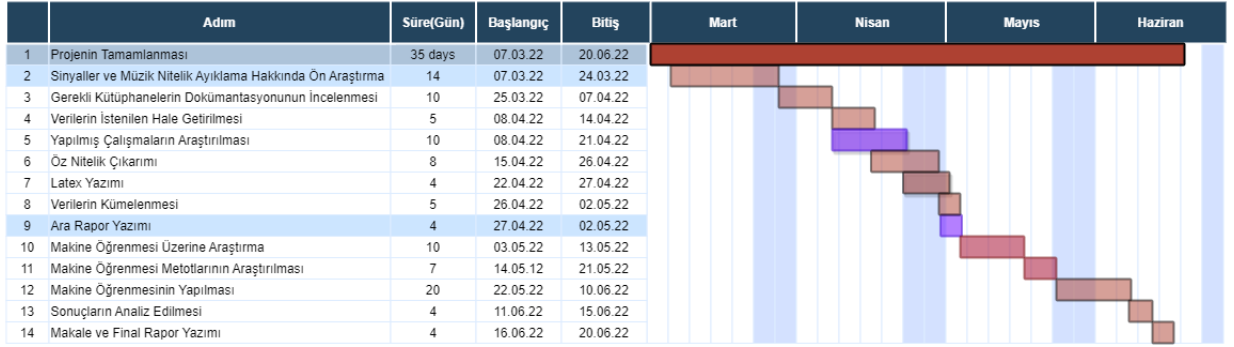
İşletim sistemi olarak bilgisayarımızda zaten bulunan Windows 10 seçilmiştir. Sebep olarak zaten bilgisayarımızda bulunması, kişisel olarak alışık olmamız ve dünyada bilgisayarlar üzerinde en çok kullanılan işletim sistemi olduğundan herhangi kullanılmak istenecek bir uygulamaya erişim olmaması çok düşük bir ihtimaldir.

Yazılım dili olarak Python, IDE olarak ise PyCharm kullanılmıştır. Veri bilimi alanında sağladığı kolaylıklar ile bilinen Jupyter Notebook'a da yine PyCharm arayüzü aracılığı ile sık sık başvurulmuştur. Veri manipülasyonu, sinyallerden öznitelik çıkarımı, makine öğrenmesi, veri görselleştirme konularında NumPy, Pandas, Librosa, SkLearn, Matplotlib isimli Python kütüphaneleri kullanılmıştır.

3.1.2 Donanım Fizibilitesi

Yazılım fizibilitesi kısmında bahsedilen veri seti ve kütüphaneleri bulunduracak kadar alanı olan ve Python yüklenebilen her işletim sistemli bilgisayar kabul edilebilir sınırlar içerisinde bulunmaktadır.

3.2 İş Gücü ve Zaman Planlaması



Şekil 3.1 İş Gücü ve Zaman Planlaması

3.3 Yasal Fizibilite

Çalışma açık kaynaklı yazılımlar kullanıldığından ve bir kar etme güdüsüyle hareket edilmediğinden yasal olarak bir sıkıntı görülmemiştir.

3.4 Ekonomik Fizibilite

Çalışmada; halka açık ücretsiz veri seti, Python kütüphaneleri kullandığından ve ayrı standartta tutulan kurumsal/şirketsel bir çalışma yapılmadığından ekonomik olarak çalışma sırasında kullandığımız kendi bilgisayarlarımız dışında ekonomik bir yük bulunmamaktadır.

4.1 Çalışmanın Hedefleri

Çalışmanın başarısı için önemli hedef adımları:

- **Ön işleme:** Veri setindeki parçaların 6'şar saniyelik uzunluklardaki eşit parçalara bölünmesi ve diğer ön işleme işlemleri.
- **Kümeleme:** Ses dosyalarına ait çıkarılan öznitelikler üzerinde çeşitli kümeleme algoritmaları uygulanarak her vektörün 2 ve 3 kümeye ayrılması.
- **Makine Öğrenmesi:** Kümelenmemiş, iki kümeye ayrılmışlar ve üç kümeye ayrılmışlar olarak düzenlenen veri setleri üzerinde makine öğrenmesi modelleri uygulanarak başarı oranları üzerinde karşılaştırmalar yapılması.

gibi söylenebilir.

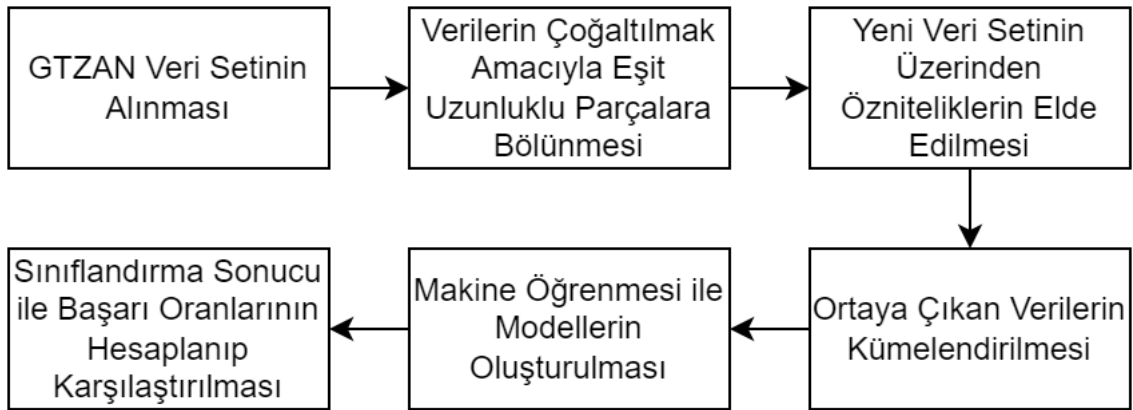
4.2 Gereksinim Analizi

- Her şeyden önce çalışma için analiz edilecek bir veri seti gerekmektedir. Bu çalışmada da incelemedeki çoğu çalışma gibi GTZAN Veri seti kullanılmaktadır. GTZAN Veri Seti'nin bu çalışma için seçilmesinin sebebi, MGR için halka açık en çok kullanılan veri seti olması ve diğer çalışmalarla karşılaştırılabilecek nitelikte sonuçlar elde etme isteğidir. Tabii ki bu veri setinin bu kadar popüler olmasının sebepleri sadece bunlar değildir, 2000'lerin başında çeşitlilik olması açısından CD'ler, radyo ve mikrofon kayıtlarıyla toplandığından olabildiğince doğal veri bulundurmaktadır [5]. Ama tabii ki hatasız da değildir. Örneklerin %7.2'si aynı kayıttan gelmekle beraber ve %5'i de birebir kopyadır. Veri setinin %10.6'sı yanlış etiketlenmiş ve bir örnek dışı bütün örneklerde bozulma bulunmaktadır [6].

- Veri seti seçildikten sonra bu verileri kolayca manipüle edebileceğimiz, öz nitelik çıkarımı yapabileceğimiz ve aynı zamanda da makine öğrenmesi için uygun olan bir dil seçilmesi gerekmektedir. Python istenilen bütün bu nitelikleri içerisindeki büyük kütüphane denizi ile kolayca karşılamaktadır. Python biraz sonra bahsedilecek kütüphaneler ile gereken iş gücü ve zamanını minimuma çekmektedir.
- Özellik çıkartmak için ise yine alanında popüler olan Librosa kütüphanesi kullanılmıştır. Librosa'yı tercih etme sebeblerimizden biri de detaylı dökümantasyonudur. İçinde bulunan örnek ve resimlerle, kullanım ve sonuç çıkarımı gayet kolaylaşmaktadır [7]. Ek olarak önde gelen veri bilimi kütüphaneleri başta olmak üzere birçok kütüphane ile uyumlu olması da büyük bir artıdır. Elde edilen vektörlerin manipülasyonu için NumPy ve Pandas kütüphaneleri kullanılmıştır. [8]. Makine öğrenmesi kısmında ise terchimiz SkLearn olmuştur.

4.3 Performans Metrikleri

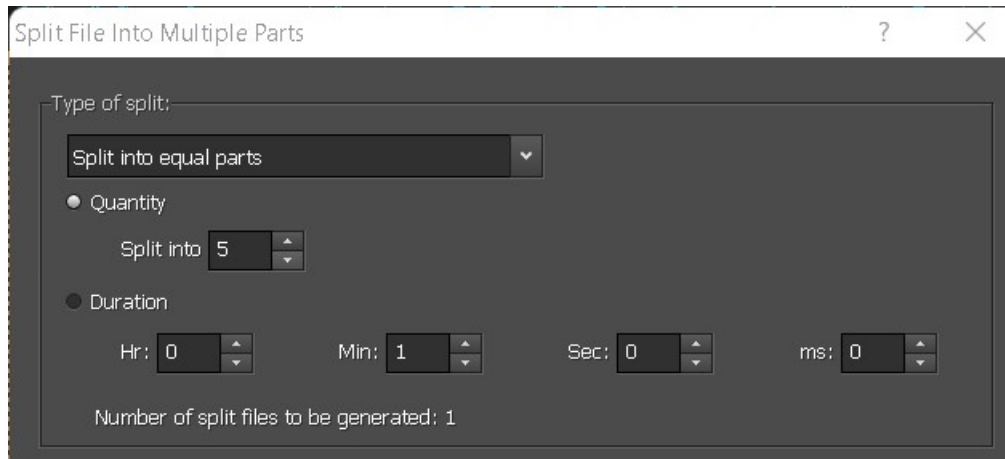
Başarı oranı Accuracy, Recall, Precision, F-Measure metrikleri üzerinden değerlendirilecektir. Model ve kümeleme yöntemiyle elde edilen küme sayısı üzerinden karşılaştırmalar yapılacaktır.



Şekil 5.1 Sistem Çalışma Akışı

5.1 Ses Dosyalarının Hazırlanması

Veri kümesindeki ses dosyaları ilgili türdeki müzik parçalarından 30 saniyelik kesitler alınarak oluşturulmuştur. Sinyal işlenecek kütüphaneye işleyebileceği boyutta dosyalar vermek ve veri setimizdeki satır sayısını arttırmak adına dosyalar 5 eşit parçaya (6 saniye) bölünmüştür.



Şekil 5.2 Ses dosyalarını bölme işlemi

5.2 Öz Niteliklerin Elde Edilmesi

1. Chroma STFT: DFT ile elde edilmiş sinyalin belli uzunluktaki bir parçasıdır. Kullanılan fonksiyonda kromagram hali çıktı olarak elde edilir [9].
2. Chroma CQT: CQT'nin oluşturulma sebebi düşük frekanslarda yaygın STFT'lerde yetersiz frekans çözünürlüğünün üstünden gelmektir. CQT, DFT gibi frekans katsayılarını hesaplar ancak bunu logaritmik ölçekte yapar. Librosa'daki versiyonu kromogram halinde çıktı verir [10].
3. MFCC: MFCC'ler bir ses sinyalinin spektral kılıfının şeklinin kompakt bir tanımı gibi düşünülebilir. MFCC'ler 1980'lerden beri daha çok konuşma tanıma da kullanılsa bile çalışmalarda müzik tespiti alanında da faydalı olduğu görülmüştür [10].
4. Spectral Center: Spektral Center, spektral enerjinin COG'si olarak tanımlanabilir. Daha detaylı ama yine de basit bir anlatımı güç spektrumunun frekans ağırlıklı toplamının ağırlıksız toplamına normalize edilmiş hali olarak söylenebilir [10].
5. Spectral Bandwidth: Spektral Yayım olarakta adlandırılır. Spektral Center etrafındaki güç spektrumunun kontrasyonu olarak anlatılabilir. Spektral Merkez etrafındaki güç spektrumunun standart sapması olarak yorumlanabilir [10].
6. Spectral Contrast: Spektrogramın her çerçevesi S tane yan banta ayrılırsa, her yan banttaki en yüksek çeyreklik ve en düşük çeyreklikteki enerjinin ortalamasının karşılaştırması Spektral Kontrast'ı verir. Yüksek kontrast değerleri açık, keskin bant sinyallerini gösterirken, düşük kontrast değerleri geniş bant seslerini işaret eder [11].
7. Spectral Rolloff: Spektral Rolloff birikmiş STFT büyüklüğünün, genel büyüklük toplamı % k miktarına erişmiş olan frekans kutucuğuna denir. " k " değeri genelde 0.85 yada 0.95 alınır [10].

Bütün öznitelikler elde edildikten sonra her bir özellik vektörüne kümeleme algoritması uygulanması aşamasına geçilmiştir. Öncelikle özellik vektörlerine kümeleme uygulanabilmesi adına flattening ve reshaping gerekli önışlemeler yapılmıştır.

5.3 Kümeleme Algoritmaları

Çalışmada kullanılacak gözetimsiz kümeleme algoritmaları:

- K-Means: "k" adet taraf ve her birinin merkezi olması hedeflenir. Önce her bir veri rastgele bir tarafa atandıktan sonra ortalama alınarak ve taraf merkezleri tekrar hesaplanarak şartı tatmin eden durum bulunur. Küme merkezleri değişmeyene kadar devam eder [12].
- GMM: Başlangıçta her veri kendi kümesidir. Her iterasyonda yakınındaki merkezi kendine katıp büyümeye devam eder. Eğer duruş şartı konmazsa kümeler sonunda birleşeceğinden tek küme oluşur [12].
- Hierarchial Clustering: İki tipi bulunmaktadır. Yukarıdan aşağıya yöntemde tek veri uzaklıklara göre parçalanarak ilerlenir. Verinin önce dendrogram yapısında olmasını gerektirdiğinden çok kullanılmaz. Aşağıdan yukarı metodunda ise GMM gibi her veri bir küme olarak kabul edilip birleştirilerek devam edilir [13].

Kümeleme algoritmaları uygulandıktan sonra elde edilen yeni özellikler üzerinde:

- Ortalama
- Standart Sapma
- Minimum
- Maksimum
- Medyan
- Mod
- Eğrilik
- Basıklık

değerleri hesaplanarak birer özellik olarak ele alınacak ve makine öğrenmesi uygulamalarına geçilecektir.

5.4 Modelleme Algoritmaları

Çalışmada kullanılacak makine öğrenmesi uygulamalarının kısaca açıklamaları aşağıdaki gibidir.:

- Logistic Regression: Verilen verileri birbirinden bağımsız olarak kabul edip, iki sonuçlu bir olayın olasılığını hesaplar [14].

- Random Forest: Verilen veri seti üzerine oluşturulan karar ağaçlarının eğitim sırasında rastgele parçalarının birleştirilmesi ile sonuçlara varma üzerine kurulu algoritmadır [15].
- K-Nearest Neighbor(KNN): Verilen "k" değerine göre, obje en yakınındaki "k" tane objeler ile ilişkilendirilir. Uzaklık öklidyen olarak hesaplanır [16].
- Support Vector Machine(SVM): Normalde lineer olarak ayıramayacak verileri daha yüksek bir boyuta çıkararak kategori edilebilecek hale getirir. Ayırıcı özellik bulunduğunda veri bu özellik çizilebilecek şekilde dönüştürülür [17].
- Gaussian Naive Bayes: Bayes Teoremi'nden oluşturulmuş olan Naive Bayes Yönteminin, verilerin sürekli ve Gauss dağılımına uygun olduğunu kabul eden varyantıdır. [18].
- XGBoost Gradient Boosting: Karar ağaçlarından gelen zayıf öğreniciyi geliştirmedir. Zayıf öğrenici rastgele olasılıktan sadece biraz yüksek başarı oranına sahip olan ağaçtır. Bu zayıf ağaç her seferinde daha da iyileştirilerek devam edilir [19].
- Stochastic Gradient Descent: İşin Lokal Minimum bulmak olan Gradient Descent'in varyantıdır. Her seferinde rastgele denilebilecek bir nokta seçildiğinden iterasyon sayısı azaltılarak performans artışı sağlanır [20].
- Decision Tree Classifier: Karar Ağaçları öncelikle en yüksek seçicilikteki niteliklerden başlayıp daha spesifiklere doğru dallar oluşturur. Verileri bu karar kavşaklarına göre sınıflandırır [21].
- XGboost random forest: Normalde gradient desteklemek için kullanılan XGBoost yöntemiyle oluşturulmuş Random Forest'tir. Aynı model temsili ve çıkarımını kullanır ama eğitim algoritması farklıdır [19].
- Gradient Boosting: Zayıf öğrenen bir modeli alıp bir sonraki en iyi model seçeneğini seçerek ilerleyen sınıflandırma yöntemidir [22].
- Adaboost: İlk uygulamalı arttırımlı algoritmadır. Başarısız sınıflandırmaları örnek alarak ve hatalarından öğrenerek sınıflandırır [23].
- Linear Support Vector Machine: SVM'lerin Lineer varyantıdır. Kerneli düz olmak zorundadır [24].

6 Uygulama

Bu bölümde çalışmada şu ana kadar elde edilmiş olan oluşturulmuş veri örnekleri yer almaktadır.

chroma_stft_1_kurtosis	chroma_cqt_1_mean	chroma_cqt_1_std_dev	chroma_cqt_1_min	...	mfcc_19_1_skew
-0.044763	0.533728	0.225491	0.098988	...	0.304363
-0.296039	0.540831	0.239653	0.097287	...	0.864507
-0.189981	0.535024	0.232014	0.128010	...	-0.132044
-0.373530	0.541433	0.230782	0.091916	...	0.138708
-0.182290	0.515297	0.228358	0.104407	...	0.763394

Şekil 6.1 Kümelenmemiş veri setinden örnek kesit

chroma_stft_1_skew	chroma_stft_1_kurtosis	chroma_stft_2_mean	chroma_stft_2_std_dev	chroma_stft_2_min
0.474243	-0.883623	0.723978	0.217329	0.412662
0.379968	-0.975335	0.753270	0.196301	0.458343
0.465004	-0.861645	0.781142	0.190635	0.479258
0.469969	-0.916342	0.765462	0.192545	0.466585
0.472148	-0.912547	0.777477	0.190409	0.483169

Şekil 6.2 GMM algoritması ile vektörleri 3 kümeye ayrılmış veri setinden örnek kesit

7 Deneysel Sonuçlar

Aşağıdaki tablolarda öznitelik vektörlerinin, kümelenmeden ve kümeleme algoritmaları ile 2 ve 3 kümelere ayrılarak oluşturdukları veri setlerine uygulanan makine öğrenmesi modellerinin; Accuracy, Recall, Precision ve F1 metrikleriyle ölçülmüş skorları, karşılaştırılarak incelenebilir.

Tablo 7.1 Herhangi bir kümeleme uygulanmadan elde edilen sonuçlar

Classifier	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.882	0.838	0.835	0.836
XGBoost	0.860	0.861	0.860	0.860
Random Forest	0.837	0.838	0.835	0.835.
Support Vector	0.803	0.802	0.802	0.801
Gradient Boosting	0.804	0.805	0.805	0.804
Linear Support Vector	0.782	0.783	0.780	0.780
Logistic Regression	0.757	0.757	0.756	0.755
XGBoost Random Forest	0.753	0.753	0.749	0.748
Linear Discriminant Analysis	0.747	0.747	0.753	0.748
Stochastic Gradient Descent	0.677	0.681	0.734	0.669
Decision Tree	0.603	0.604	0.605	0.603
Gaussian Naive Bayes	0.546	0.546	0.550	0.528
AdaBoost	0.354	0.353	0.358	0.301

Tablo 7.2 K-Means Clustering algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar

Classifier	Accuracy	Recall	Precision	F1
K Neighbors	0.869	0.869	0.871	0.869
XGBoost	0.844	0.845	0.844	0.843
Random Forest	0.829	0.829	0.829	0.826
Support Vector Machine	0.775	0.777	0.774	0.773
Gradient Boosting	0.797	0.797	0.798	0.796
Linear Support Vector	0.783	0.784	0.781	0.781
Logistic Regression	0.750	0.751	0.748	0.747
XGBoost Random Forest	0.738	0.739	0.736	0.732
Linear Discriminant Analysis	0.737	0.737	0.742	0.738
Stochastic Gradient Descent	0.670	0.668	0.731	0.654
Decision Tree	0.568	0.567	0.572	0.568
Gaussian Naive Bayes	0.528	0.527	0.543	0.505
AdaBoost	0.351	0.351	0.319	0.302

Tablo 7.3 K-Means Clustering algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar

<i>Classifier</i>	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.695	0.695	0.698	0.693
XGBoost	0.833	0.835	0.833	0.832
Random Forest	0.810	0.810	0.808	0.806
Support Vector	0.747	0.748	0.746	0.745
Gradient Boosting	0.794	0.795	0.795	0.794
Linear Support Vector	0.751	0.751	0.749	0.749
Logistic Regression	0.738	0.738	0.736	0.735
XGBoost Random Forest	0.735	0.736	0.735	0.731
Linear Discriminant Analysis	0.739	0.739	0.743	0.739
Stochastic Gradient Descent	0.639	0.638	0.714	0.623
Decision Tree	0.571	0.570	0.570	0.569
Gaussian Naive Bayes	0.520	0.520	0.537	0.498
AdaBoost	0.354	0.358	0.361	0.310

Tablo 7.4 GMM algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar

<i>Classifier</i>	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.796	0.796	0.799	0.795
XGBoost	0.823	0.823	0.822	0.821
Random Forest	0.805	0.805	0.804	0.801
Support Vector	0.751	0.752	0.750	0.748
Gradient Boosting	0.773	0.774	0.775	0.773
Linear Support Vector	0.727	0.728	0.724	0.724
Logistic Regression	0.712	0.713	0.711	0.710
XGBoost Random Forest	0.722	0.723	0.718	0.717
Linear Discriminant Analysis	0.714	0.714	0.717	0.714
Stochastic Gradient Descent	0.614	0.614	0.711	0.602
Decision Tree	0.551	0.552	0.554	0.551
Gaussian Naive Bayes	0.522	0.523	0.534	0.503
AdaBoost	0.325	0.327	0.311	0.265

Tablo 7.5 GMM algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar

<i>Classifier</i>	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.642	0.642	0.648	0.640
XGBoost	0.818	0.819	0.817	0.817
Random Forest	0.779	0.780	0.779	0.776
Support Vector	0.716	0.717	0.715	0.713
Gradient Boosting	0.772	0.773	0.774	0.772
Linear Support Vector	0.715	0.716	0.712	0.712
Logistic Regression	0.7	0.701	0.699	0.698
XGBoost Random Forest	0.716	0.716	0.712	0.711
Linear Discriminant Analysis	0.713	0.713	0.718	0.714
Stochastic Gradient Descent	0.702	0.604	0.682	0.593
Decision Tree	0.543	0.543	0.546	0.543
Gaussian Naive Bayes	0.514	0.514	0.523	0.495
AdaBoost	0.283	0.286	0.229	0.221

Tablo 7.6 Hierarchical Clustering algoritması kullanılarak vektörlerin 2 kümeye ayrılmasıyla elde edilen sonuçlar

<i>Classifier</i>	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.753	0.754	0.754	0.751
XGBoost	0.813	0.813	0.812	0.811
Random Forest	0.770	0.771	0.768	0.766
Support Vector	0.744	0.743	0.742	0.740
Gradient Boosting	0.763	0.764	0.763	0.762
Linear Support Vector	0.742	0.743	0.739	0.739
Logistic Regression	0.721	0.721	0.718	0.718
XGBoost Random Forest	0.704	0.704	0.703	0.697
Linear Discriminant Analysis	0.721	0.721	0.723	0.721
Stochastic Gradient Descent	0.648	0.648	0.724	0.641
Decision Tree	0.531	0.532	0.533	0.531
Gaussian Naive Bayes	0.505	0.505	0.512	0.480
AdaBoost	0.320	0.319	0.318	0.248

Tablo 7.7 Hierarchical Clustering algoritması kullanılarak vektörlerin 3 kümeye ayrılmasıyla elde edilen sonuçlar

<i>Classifier</i>	Accuracy	Recall	Precision	F1
K Nearest Neighbors	0.584	0.584	0.586	0.579
XGBoost	0.798	0.797	0.796	0.795
Random Forest	0.762	0.763	0.759	0.756
Support Vector	0.717	0.718	0.715	0.713
Gradient Boosting	0.756	0.756	0.758	0.756
Linear Support Vector	0.731	0.732	0.727	0.728
Logistic Regression	0.717	0.717	0.714	0.713
XGBoost Random Forest	0.707	0.709	0.707	0.701
Linear Discriminant Analysis	0.715	0.717	0.719	0.716
Stochastic Gradient Descent	0.605	0.603	0.716	0.597
Decision Tree	0.531	0.531	0.533	0.531
Gaussian Naive Bayes	0.503	0.503	0.509	0.476
AdaBoost	0.297	0.302	0.266	0.242

Skor tablolarındaki sonuçlara göre, araştırılan konu doğrultusunda genel bir karşılaştırma yapılabilmesi adına Accuracy metriği baz alınarak en başarılı 5 model üzerinden hazırlanan karşılaştırma tablosu Tablo 7.8’de görülebilir.

Tablo 7.8 Genel karşılaştırma tablosu

	KNN	XGBoost	Random Forest	Support Vector	Gradient Boosting
No Cluster	0.882	0.860	0.837	0.803	0.804
K-Means -2	0.869	0.844	0.829	0.775	0.797
K-Means -3	0.695	0.833	0.810	0.747	0.794
GMM -2	0.796	0.823	0.805	0.751	0.773
GMM -3	0.642	0.818	0.779	0.716	0.772
Hierarchical -2	0.753	0.813	0.770	0.744	0.763
Hierarchical -3	0.584	0.798	0.762	0.717	0.756

8 Performans Analizi

Sinyal işleme sonucunda çıkartılan öznitelik vektörlerine herhangi bir kümeleme işlemi uygulanmadan elde edilen başarı oranlarının K-Nearest Neighbours, Extreme Gradient Boosting, Random Forest, Support Vector ve Gradient Boosting sınıflandırma metotları ile %80'in üzerine çıktığı gözlenmiştir.

Deney sonuçlarının incelenmesine gelmeden önce değerlendirilmesi gereken bir diğer konu hangi metrikler üzerinden inceleme yapılacağıdır. Araştırmada ölçüm metrikleri olarak Accuracy, Recall, Precision ve F1 kullanılmış fakat aralarında kayda değer farklar gözlenmemiştir. Veri setinin niteliği de göz önüne alındığında genel karşılaştırmanın Accuracy metriği üzerinden yapılmasına karar verilmiştir.

En başarılı 5 model üzerinden Accuracy metriği baz alınarak karşılaştırmalar yapılan Tablo 7.8 incelendiğinde yöntem ve sayı farketmeksizin öznitelik vektörleri üzerinde kümeleme uygulamanın başarı skorlarını düşürdüğü görülmüştür. Yöntemler kendi aralarında karşılaştırıldığında ise en az düşüş K-Means ile yaşanırken en dramatik düşüşün Hierarchical Clustering yönteminde olduğu görülmektedir. Ayrıca küme sayısının etkisinin incelenmesi istendiğinde, küme sayısı arttıkça doğruluk skorunda oldukça dramatik bir düşüşün olduğu gözlemlenmektedir.

Kümeleme yapılmadığında en yüksek sonucu veren KNN yönteminin en az K-Means , en çok Hierarchial Clustering'den etkilendiği görülmüştür. Kümeleme olmadığı hale kıyasla Hier 2'li kümeleme %10 daha az başarı göstermiştir. Kümeleme algoritmaları 3'lü halde yapıldığında ise KNN'in başarısının %15'ten fazla azaldığı gözlemlenmiştir, Hier-3'te ise kümesiz hale kıyasla neredeyse %30 başarı düşüşü olmuştur. Kmeans-2 de ise Accuracy düşmüş olsa da diğer bütün metrikler %3 kadar yükselmiştir.

Decision Tree, LDA modellerinde Kümeleme Yöntemine göre puanları düşse de küme sayısının artması başarısını değiştirmemiştir. AdaBoost'ta GMM ile kümeleme sayısı artınca neredeyse KNN kadar bir başarı düşüşü olup %30 başarının altına inmiştir.

9 Sonuç

Projenin amacı sayısal işaret işleme yöntemleri ile müzik türü sınıflandırması alanında, öznitelik vektörlerinin üzerinde kümeleme algoritmaları ile kümele yapılmasının başarı oranlarına etkisini araştırmaktır. Projede hazır öznitelik veri seti kullanılmayıp, '.wav' uzantılı müzik dosyalarından Librosa yardımı ile özellik çıkarımı yapılmıştır. Özellik çıkarımı yapılması sırasında vektörler elde edildiği anda K-Means, Gmm ve Hierarchical kümeleme yöntemleri ile 2 ve 3 küme olmak üzere kümelere ayrılmış bu şekilde 7 adet (No cluster, K-Means 2, K-Means 3, Gmm 2, Gmm 3, Hierarchical 2, Hierarchical 3) veri seti elde edilmiştir. Elde edilen 7 veri seti ayrı ayrı 14 farklı makine öğrenmesi modeline beslenmiş ve sonuçlar çeşitli metrikler ile karşılaştırılmıştır. Metrikler arasında kayda değer farkların olmadığı, Accuracy metriğinin ilgili sınıflandırma özelinde yeterince anlamlı olduğu görülmüştür. Sayısal veriler kullanılarak yapılan karşılaştırmalar 7. Bölüm'de yer almakta olup bu bölümde sonuçlar tartışılacaktır.

Nihai sonuç olarak, bahsedilen şekilde yapılan uygulamaların ardından yöntem ve küme sayısı farketmeksizin öznitelik vektörleri üzerinde kümeleme uygulamanın başarı skorlarını düşürdüğü görülmüştür. En az negatif etkinin K-Means ile yapılan kümelemeden geldiği görülürken en şiddetli negatif etkinin Hierarchical kümeleme yönteminden geldiği görülmüştür. Ayrıca küme sayısının artması da birçok yöntemde negatif etkiyi arttırmıştır. KNN sınıflandırma metodu, kümeleme uygulanmadan en başarılı sonuçları veren metot olmasına rağmen kümelemeden en dramatik şekilde etkilenen metot olmuştur. Kullanılan 14 modelde de düşüş yaşanması ise; kaybın, model - kümeleme yöntemi veya model - küme sayısı ilişkisinden değil, uygulanan yöntem veya uygulama biçiminden kaynaklı olduğunu göstermiştir.

- [1] C. Xu, N. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 5, 2003, pp. V-429. DOI: 10.1109/ICASSP.2003.1199998.
- [2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, 2002. DOI: 10.1109/TSA.2002.800560.
- [3] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," Jan. 2003, pp. 282-289. DOI: 10.1145/860435.860487.
- [4] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303-319, 2011. DOI: 10.1109/TMM.2010.2098858.
- [5] A. Olteanu. "Gtzan dataset - music genre classification." (2020), [Online]. Available: <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>.
- [6] B. L. Sturm, "An analysis of the gtzan music genre dataset," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012, pp. 7-12.
- [7] "Librosa:0.9.1." (Feb. 2022), [Online]. Available: <https://zenodo.org/record/6097378>.
- [8] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: <https://doi.org/10.5281/zenodo.3509134>.
- [9] D. Ellis, "Chroma feature analysis and synthesis," *Resources of Laboratory for the Recognition and Organization of Speech and Audio-LabROSA*, vol. 5, 2007.
- [10] A. Lerch, *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press, 2012.
- [11] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings. IEEE International Conference on Multimedia and Expo*, IEEE, vol. 1, 2002, pp. 113-116.
- [12] S. R. A. Ahmed, I. Al Barazanchi, Z. A. Jaaz, and H. R. Abdulshaheed, "Clustering algorithms subjected to k-mean and gaussian mixture model on multidimensional data set," *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 448-457, 2019.

- [13] F. Nielsen, “Hierarchical clustering,” in *Introduction to HPC with MPI for Data Science*. Cham: Springer International Publishing, 2016, pp. 195–211, ISBN: 978-3-319-21903-5. DOI: 10.1007/978-3-319-21903-5_8. [Online]. Available: https://doi.org/10.1007/978-3-319-21903-5_8.
- [14] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [15] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [16] Z. Zhang, “Introduction to machine learning: K-nearest neighbors,” *Annals of translational medicine*, vol. 4, no. 11, 2016.
- [17] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [18] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, “Fast gaussian naive bayes for searchlight classification analysis,” *Neuroimage*, vol. 163, pp. 471–479, 2017.
- [19] T. Chen *et al.*, “Xgboost: Extreme gradient boosting,” *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.
- [20] L. Bottou, “Stochastic gradient descent tricks,” in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 421–436.
- [21] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [22] A. J. Izenman, “Linear discriminant analysis,” in *Modern multivariate statistical techniques*, Springer, 2013, pp. 237–280.
- [23] R. E. Schapire, “Explaining adaboost,” in *Empirical inference*, Springer, 2013, pp. 37–52.
- [24] S. Suthaharan, “Support vector machine,” in *Machine learning models and algorithms for big data classification*, Springer, 2016, pp. 207–235.

BİRİNCİ ÜYE

İsim-Soyisim: Tarık AYTEK
Doğum Tarihi ve Yeri: 26.06.1999, İstanbul
E-mail: tarik.aytek@std.yildiz.edu.tr
Telefon: 0545 760 19 99
Staj Tecrübeleri:

İKİNCİ ÜYE

İsim-Soyisim: Tuğrul Alp ÖZBUCAK
Doğum Tarihi ve Yeri: 01.02.2000, Samsun
E-mail: alp.ozbucak@std.yildiz.edu.tr
Telefon: 0507 978 56 28
Staj Tecrübeleri: NTT DATA Business Solutions Turkey Analytics Products

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python 3
Gerekli RAM: 2GB
Gerekli Disk: 256MB